Convex-Based Thermal Management for 3D MPSoCs Using DVFS and Variable-Flow Liquid Cooling

Francesco Zanini[1], David Atienza[2], and Giovanni De Micheli[1]

[1] Integrated Systems Laboratory (LSI), EPFL, Lausanne, Switzerland
[2] Embedded Systems Laboratory (ESL), EPFL, Lausanne, Switzerland
name.surname@epfl.ch

Abstract. In this work, we propose a novel online thermal management approach based on *model predictive control* for 3D multi-processors system on chip (*MPSoCs*) using microfluidic cooling. The controller uses dynamic voltage and frequency scaling (*DVFS*) for the computational cores and adjusts the liquid flow rate to meet the desired performance requirements and to minimize the overall MPSoC energy consumption (MPSoC power consumption+cooling power consumption). Our experimental results illustrate that our policy satisfies performance requirements and maintains the temperature below the specified threshold, while reducing cooling energy by up to 50% compared with traditional state-of-the-art liquid cooling techniques. The proposed policy also keeps the thermal profile up to $18^{\circ}C$ lower compared with state of the art 3D thermal management using variable-flow liquid cooling.

1 Introduction

Power and thermal management are important challenges for multicore systems. Since power density is increasing, heat extraction is becoming more difficult. Moreover, in 3D stacked chips it is even more complex to develop efficient cooling mechanisms. However, liquid cooling has emerged as a potential solution to address the high temperatures in 3D chips [3], due to the higher heat removal capability of liquids in comparison to air. Liquid cooling is performed by attaching a cold plate with built-in microchannels, and/or by fabricating microchannels in the silicon layers of the 3D-MPSoC architectures. Then, a coolant fluid is pumped through the microchannels to remove the heat. The flow rate of the pumps can be altered dynamically, and the pump power consumption increases quadratically with the increase in flow rate [3]. Thus its contribution to the overall system energy is not negligible [16], and new thermal management policies must be developed to exploit this new cooling technology while considering the pumping power overhead.

The main contribution of this work is a novel thermal management approach for 3D stacks that controls both DVFS and a variable-flow liquid cooling using convex optimization to meet the desired performance and minimal energy requirements. The optimization process is applied at run-time using the convex-solver proposed by [14]. At this stage the convex solver finds the optimum frequency assignment for the inputs of the MPSoC system that will maximize performance under temperature constraints.

We perform experiments on a 3D multicore architecture case study based on Niagara T1 UltraSparc2 cores [4] using benchmarks ranging from web-accessing to playing multimedia. Results show that the proposed method guarantees that scenarios with dangerous thermal profiles are avoided while satisfying the application performance requirements. Moreover, cooling energy is reduced by up to 50% compared with state of the art liquid cooling policies. In addition, the proposed policy keeps the average thermal profile up to $18°C$ lower compared with state of the art polices using variable-flow liquid cooling, like [16].

2 Related Work

Accurate thermal modeling of liquid cooling is critical in the design and evaluation of systems and policies. HotSpot [5] is a thermal model tool that calculates transient temperature response given the physical and power consumption characteristics of the chip. The latest versions of HotSpot include 3D modeling capabilities and liquid-cooled systems as well [6]. Finally, 3D-ICE [7] is a new thermal modeling tool specifically designed for 3D stacks, and includes inter-layer liquid cooling modeling capabilities.

Many researchers in computer architecture have recently focused on thermal control for *Multi-Processor System on Chips* (MPSoCs) [11], [8]. Processor power optimization and balancing using DVFS have been proposed in several works [8], [19]. However in all aforementioned policies there is not a guarantee to avoid hotspots by performing this optimization, because the policy targets power optimization and not hotspot avoidance.

More advanced solutions apply the concepts of model-predictive control to turn the control from open-loop to closed-loop [9], [10]. In [18] a similar concept is tailored for multi-modal video sensor nodes. In [15] a convex optimization-based approach is presented. The advantage of our technique over these methods is the new degree of freedom given by active liquid cooling. In [1] and [16], thermal management methods for 3D MPSoCs using a variable-flow liquid cooling have been proposed. These policies use simple heuristics to control the temperature profile of the 3D MPSoC, so there is not formal guarantee of optimality using this approach.

3 Modeling 3D Systems with Liquid Cooling

This paper deals with 3D MPSoCs stacking two or more dies. As an example, Figure 1(a),(b),(c) shows a 3D system consisting of 4-tiers. There are four silicon layers (A, B, C, D) (with various functional units grouped into p islands with independent clock frequency and voltage supplies), where *microchannels* are etched in silicon bulk for liquid cooling. The model abstracts the interconnect on chip as copper layers (A, B, C, D). For every silicon layer there is a total of nc linear microchannels $Ch_1 \dots Ch_{nc}$. Microchannels are assumed to be equal in dimensions and a uniform coolant flux is assumed in channels of the same layer. All microchannels belonging to the same layer are connected to a pump. In the model shown in Figure 1(c) there is a total of 4 pumps connected to the microchannels of the 4 silicon layers. Fluid flows through channels belonging to different layers with different flow rates, according to the power of each pump. The liquid flow rate provided by each pump can be dynamically altered at runtime.

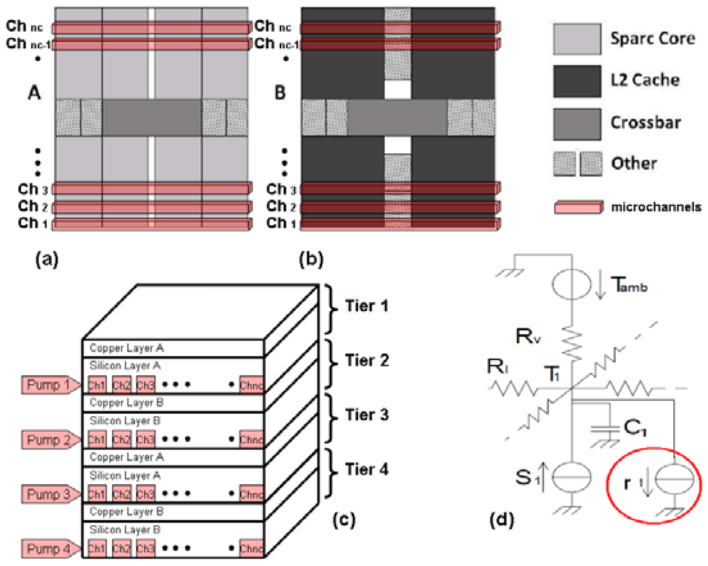


Fig. 1. 3-D stacked MPSoC with liquid cooling: (a)silicon layer type-A, (b)silicon layer type-B, (c)overall MPSoC view, (d)resistive network model

3.1 3D Heat Propagation Model

We model heat propagation in the 3D stacks using a network of thermal resistances and capacitances. To model the architecture shown in Figure 1(a),(b),(c) we propose an extension of the model presented in [15]. In particular, the active cooling (for cell i) is modeled by a current sink r_i, as highlighted by the circle in Figure 1(d). This current sink models the capability of the cooling system to remove heat in a specific location of the MPSoC. Following [15], we model the heat propagation process as:

$$\mathbf{t_{\tau+1}} = \mathbf{A}\mathbf{t}_\tau + \mathbf{B}\mathbf{p}_\tau \tag{1}$$

$$\tilde{\mathbf{t}}_\tau = \mathbf{C}\mathbf{t}_\tau \tag{2}$$

We assume that the total number of cells in all layers of the 3D MPSoC structure is n, the total number of cores is p and the total number of pumps is z. Matrices $\mathbf{A} \,\epsilon\, \Re^{\mathbf{n\times n}}$ and $\mathbf{B} \,\epsilon\, \Re^{\mathbf{n\times(p+z)}}$ describe the heat propagation properties of the MPSoC. At time τ, the temperature of the next simulation step of cell i, i.e. $(\mathbf{t_{\tau+1}})_\mathbf{i}$ can be computed thanks to Equation 1. The vector $\mathbf{p} \,\epsilon\, \Re^{\mathbf{p+z}}$ is the power input vector. The first p entries are the normalized power consumption for each of the p frequency islands (cores), while the remaining z entries are the normalized cooling power for each of the z pumps.

The relation between the frequency assignment at time τ, $\mathbf{f}_\tau \,\epsilon\, \Re^{\mathbf{P}}$, and the power consumption is assumed to be quadratic [5]. Equation 2 describes the choice of temperature sensors inside the MPSoC. Matrix $\mathbf{C} \,\epsilon\, \Re^{\mathbf{s\times n}}$ relates the temperature value of each cell to the temperature measurement of a particular sensor location in order to represent realistic IC designs that can only contain a discrete number of s temperature sensors. The law that relates the microchannel flow-rate to heat extraction has been taken from [7]. Thus, we consider that the amount of heat r_i extracted in cell i by the fluid in the microchannel controlled by pump j can be approximated by: $r_i = m_j \cdot \gamma_{i,j} \cdot (t_i - t_{fluid})$ where the fluid temperature is t_{fluid}, t_i is the temperature of cell i and $\gamma_{i,j}$ is the constant modeling the channel heat extraction properties. Vector $\mathbf{m} \in \Re^{\mathbf{z}}$ is the normalized amount of heat that can be extracted for each of the z independent pumps. Hence, by varying vector $\mathbf{m}$, the cooling power (flow rate of the cooling liquid) is varied to achieve the desired heat extraction. In our model, we used the temperature mapping from [7] to derive $\gamma_{i,j}$. Experiments have shown that by updating $\gamma_{i,j}$ every time the policy is applied ($10ms$ in our simulation setup), our approximation leads to a maximum error up to $\pm 5\%$.

3.2 Workload Model

The workload is an abstraction of what the operating system generates from higher-level software layers. For each p clock islands (cores), the workload is defined as the minimum value of the clock frequency that the functional unit should have to execute the required tasks within the specified system constraints.

The workload requirement at time τ is defined as a vector $\mathbf{w}_\tau \in \Re^{\mathbf{p}}$, where $(\mathbf{w}_\tau)_\mathbf{i}$ is the workload requirement value for input i at time τ. $(\mathbf{w}_\tau)_\mathbf{i}$ is the frequency that cores associated with input i from time τ to time $\tau + 1$ should have in order to satisfy the desired performance requirement coming from the scheduler.

We assume a continuous control on the frequency ranging from the minimum frequency possible by each core ($\mathbf{f}_{\min}$) to their maximum frequency value ($\mathbf{f}_{\max}$), namely:

$$\mathbf{f}_{\min} \preceq \mathbf{w}_\tau \preceq \mathbf{f}_{\max} \;\; \forall \, \tau \tag{3}$$

When $(\mathbf{w}_\tau)_\mathbf{i} > (\mathbf{f}_\tau)_\mathbf{i}$, the workload cannot be processed and so it needs to be stored and rescheduled in the following clock cycles. The way we measure the performance of the system in achieving the requested workload requirements at time τ is given by the vector $\mathbf{u}_\tau \in \Re^{\mathbf{p}}$ as follows:

$$\mathbf{u}_\tau = \mathbf{w}_\tau - \mathbf{f}_\tau \tag{4}$$

Therefore, we call $\mathbf{u}_\tau$ the undone workload at time τ, which expresses the difference at time τ between the requested workload and the actual one executed by the MPSoC.

4 Policy Computation

The proposed thermal management approach uses both DVFS and variable-flow liquid cooling to meet the desired requirements, which are represented by a two-term cost function. The first one is related to power minimization (3D-MPSoC power consumption and liquid cooling pumping system power consumption) and the second one to the performance loss (undone work). The solution of following minimization are the

3D MPSoC frequencies and cooling pumps speeds necessary to meet the desires requirements. The control problem is formulated as the following convex optimization problem:

$$
\begin{aligned}
J = & \sum_{\tau=1}^{h} \left(\|\mathbf{R}\mathbf{p}_\tau\|_\mathbf{j} + \|\mathbf{T}\mathbf{u}_\tau\|_\mathbf{b} \right) && (5)\\
min\ J & && (6)\\
subject\, to: \ & \mathbf{f}_{\min} \preceq \mathbf{f}_\tau \preceq \mathbf{f}_{\max}\ \forall\,\tau && (7)\\
& \mathbf{x}_{\tau+\mathbf{1}} = \mathbf{A}\mathbf{x}_\tau + \mathbf{B}\mathbf{p}_\tau\ \forall\,\tau && (8)\\
& \tilde{\mathbf{C}}\mathbf{x}_{\tau+\mathbf{1}} \preceq \mathbf{t}_{\max}\ \forall\,\tau && (9)\\
& \mathbf{u}_\tau \succeq \mathbf{0}\ \forall\,\tau && (10)\\
& \mathbf{u}_\tau = \mathbf{w}_\tau - \mathbf{f}_\tau\ \forall\,\tau && (11)\\
& \mathbf{l}_\tau \succeq \mu\mathbf{f}_\tau^\alpha\ \forall\,\tau && (12)\\
& -\mathbf{w} \preceq \mathbf{m}_{\tau+\mathbf{1}} - \mathbf{m}_\tau \preceq \mathbf{w}\ \forall\,\tau && (13)\\
& 0 \preceq \mathbf{m}_\tau \preceq \mathbf{1}\ \forall\,\tau && (14)\\
& \mathbf{p}_\tau = [\mathbf{l}_\tau; \mathbf{m}_\tau]\ \forall\,\tau && (15)
\end{aligned}
$$

It is important to highlight that the matrices $\mathbf{A}$, $\mathbf{B}$ used in previous equations are constant during the h time steps the system tries to minimize the cost function J, and are then updated every time the policy is applied. In our optimization problem formulation, h is the time horizon [9](or number of time steps) to minimize the cost function J. Then, matrices A, B are constant during these next h time steps, and are then updated every time the predictive policy is applied.

Function J is expressed by a sum where the summation index τ ranges from 1 to h. The first term $\|\mathbf{R}\mathbf{p}_\tau\|_\mathbf{j}$ is the j norm (in our implementation $j = 1$) of the power input vector p weighted by matrix $\mathbf{R}$. Power consumption is generated here by two main sources: the voltage-frequency setting of the 3D MPSoC and the liquid cooling pumping power. Vector p is a vector containing normalized power consumption data of both the cores and the cooling pumps. Matrix $\mathbf{R}$ contains the maximum value of the power consumption of both the cores (first p diagonal entries) and the cooling pumps (last z diagonal entries).

The second term $\|\mathbf{T}\mathbf{u}_\tau\|_\mathbf{b}$ is the b norm (in our implementation $b = 1$) of the amount of predicted required workload that has not been executed. The weight matrix $\mathbf{T}$ quantifies the importance that executing the workload required from the scheduler has in the optimization process.

Inequality 7 defines the range of working frequencies that can be used. It enables a continuous range of frequency settings but this does not prevent from adding in the optimization problem a limitation on the number of allowed frequency values. Equation 8 defines the evolution of the system according to the present state and inputs. Equation 9 states that temperature constraints should be respected at all times and in all specified locations. Since the system cannot execute jobs that have not arrived, every entry of $\mathbf{u}_\tau$ has to be greater than or equal to 0 as stated by Equation 10. The undone work at time τ, u_τ is defined by Equation 11. Equation 12 defines the relation between the power vector $\mathbf{l}$ and the working frequencies. μ is a technology-dependent constant. Because of the fact that all constraints in the minimization problem must be convex functions, we relaxed the original power equation to the convex inequality of Equation 12. By doing this operation we changed the original minimization problem to the problem described by the convex Equations 5-12. It can be shown that the resulting relaxed convex problem is equivalent to the original problem with the equality constraint [20].

Equation 15 defines formally the structure of vector $\mathbf{p}$ as described in Section 3.1. Vector $\mathbf{l} \in \Re^{\mathbf{p}}$ is the power input vector, where p is the number of frequency islands composing the 3D-MPSoC. Vector $\mathbf{m} \in \Re^{\mathbf{z}}$ contains the normalized amount of cooling power for each of the z independent pumps. Equations 13-14 define constraints on the liquid cooling management. Equation 14 states that $\mathbf{m}$ is a normalized value and it can range from 0 to 1. Equation 13 defines the maximum increment/decrement that the normalized pump can have between two consequent applications of the policy. In other terms this value takes into account the mechanical time dynamics of the pump. Their values are stored in vector $\mathbf{w} \in \Re^{\mathbf{z}}$.

The result of the optimization is an optimal sequence of future control moves (i.e., frequency settings for the cores of the 3D MPSoC which are stored in vector $\mathbf{f}$). To increase the performance of our proposed policy, history information about the task arrival process are exploited by the proposed algorithm. Matrix $\mathbf{T}$ is chosen accordingly to the reliability of the workload prediction. We have selected these parameters to achieve a good prediction, according to empirical studies performed on different benchmarks [12].

5 Experimental Setup

5.1 3D MPSoC Model

The MPSoC structure we are considering is presented in Figure 1(a),(b),(c). The floorplan has been modelled using technological parameters and coefficients taken from [1] and [4]. This structure has a maximum operating frequency of 1.2 GHz and the maximum power consumption of each core at this frequency is 5 W.

To implement the voltage and frequency scaling techniques, we use frequencies ranging from $\mathbf{f}_{\min}$ to 1.2GHz, see [4] for details. In this range, only specific values of frequencies are allowed. These values are generated from the integer division of the maximum clock frequency by scaling factors as proposed in [17].

We compute the leakage power of processing cores as a function of their area and the temperature. We assume a base leakage power density of $0.25Wmm^2$ at $383°K$ for $90nm$, as in [19]. T_o accounts for the temperature effects on leakage power and we use the model proposed in [1]. In this case, the leakage power at a temperature T_o $°K$ is given by: $P(T) = Po \cdot e^{\beta(T-383)}$, where P_o is the leakage power at $383°K$, and β is a technology dependent coefficient. Finally we set $\beta = 0.017$ [16].

The number of independent flow rates is 4 and the spacing between two microchannels on the same layer is $100\mu m$. We assume that a pump connected to all microchannels of the same layer, such as a centrifugal pump [16], is responsible for the fluid injection to the whole system. This pump has the capability of producing large discharge rates at small pressure heads. Liquid is injected to the stacks from this pump via a pumping network. To enable using different flow rates for each of the 4 stacks, the cooling infrastructure includes valves in the network. Cooling microchannels parameters and cooling pump power consumption values are taken from [16].

5.2 Policy Setup

According to the general model of Equations 5-12, the problem formulation is the following. Matrix $\mathbf{T}$ is set to be an identity matrix while matrix $\mathbf{R}$ contains the maximum value of the power consumption of both the cores and the cooling pumps (power values from [1]). The policy minimizes the sum of all contributions to the 3D MPSoC power consumption as well as the undone workload. For this reason, we set both the norms b and j to 1.

All the others constraints expressed by Equations 7-12 are considered inside the problem formulation. The policy is applied every $T_{pol} = 10ms$, while the simulation step for the discrete time integration of the RC thermal model has been set to 200μs. The maximum temperature limit is set to $370°K$. The room temperature and t_{fluid} are set to $300°K$. In the problem formulation, we used $\alpha = 2$ to establish the relation between the frequency setting and the power consumption. The linear predictor has been designed using a 3^{rd} order polynomial equation, an observation window of $600ms$ and a prediction length equal to $50ms$ in the future. The optimization process is done online using the convex solver proposed in [14]. These operations, have been performed on standard processors (i.e., Core Duo @ 2GHz) in few tenth of microseconds. This time is 3 orders of magnitude smaller compared with the time the policy is applied (i.e.10ms). The time constants needed by the mechanical dynamics of the cooling pumps to go from 0 to maximum power is set to $400ms$.

We use workload traces collected from real applications running on an UltraSPARC T1. We record the utilization percentage for each hardware thread at every second using *mpstat* for several minutes for each benchmark. We use various real-life benchmarks including web server, database management, and multimedia processing.

6 Experimental Results

In our experiments, we compare the proposed 3D thermal management method with state-of-the-art thermal management techniques based on DVFS, load balancing and variable flow liquid cooling [1], [16], [11], [13], [2].

Dynamic load balancing ($\mathbf{LB}$) [11] balances the workload by moving threads from a core's queue to another if the difference in queue lengths is over a threshold. Temperature-triggered task migration ($\mathbf{TTTM}$) [13] moves tasks from a core if that core exceeds the threshold temperature. TTTM has an impact on performance resulting from the time overhead required to move tasks between the cores (e.g., context switch overhead and cold start effects). In this work we assume a $1ms$ overhead when a thread is migrated to a new core [1], [2]. For previously mentioned polices, if the temperature goes higher than $420°K$, the system shuts down until the maximum MPSoC temperature returns below $250°K$. In temperature triggered DVFS ($\mathbf{TTDVFS}$) [11] the voltage and frequency settings are reduced to the 10% of the maximum value when the maximum MPSoC temperature exceed the threshold value set to $370°K$. TTTM and TTDVFS can also be combined into a joint policy called ($\mathbf{TTTM\_TTDVFS}$) [2].

We experiment with both air-cooled ($\mathbf{AC}$) and liquid-cooled ($\mathbf{LC}$) systems for comparison purposes. In $\mathbf{LC\_LB}$, we apply 100% of the maximum flow rate (0.0323 l/min per cavity [16]).We also consider in the comparison the latest state-of-the-art liquid cooling methods proposed in proposed in [1] and [16]. These methods employ a variable-flow liquid cooling combined with DVFS. We refer to the first method as $\mathbf{LC\_VF}$ and to the second one as $\mathbf{LC\_Fuzzy}$.

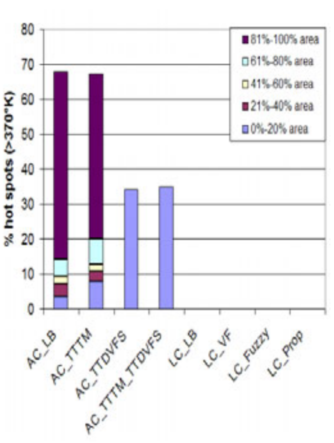
Thermal impact of all the policies on the system is shown in Figure 2(a). This figure compares the percentage of run-time execution where the maximum MPSoC temperature is higher than $370°K$. The hotspot area is labeled as the overall MPSoC area percentage.

The first four policies are air cooled methods, while the last four are liquid cooled. As Figure 2(a) shows, the first ones are not able to avoid hot spots. $\mathbf{AC\_LB}$ and $\mathbf{AC\_TTTM}$ present hot spots for up to 67% of the execution time, and in addition to that, these hot spots affect more than 80% of the total MPSoC area. Methods using temperature-triggered DVFS show a better performance. This is shown for $\mathbf{AC\_TTDVFS}$ and $\mathbf{AC\_TTTM\_TTDVFS}$.

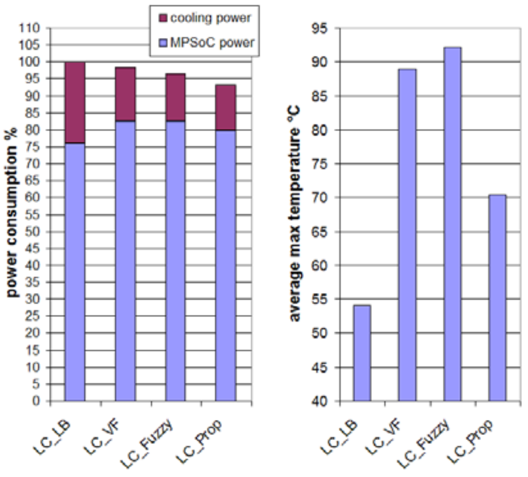
Hence, they present hot spots for only 34% and 35% of the execution time, respectively. In addition, these hot spots cover less than 20% of the overall MPSoC area.

Nevertheless, overall air cooled policies do not completely avoid hot spots. The reason is because the 4-tier stacked architecture is unable to dissipate the heat of inner layers by using only a heat spreader. These results indicate the benefits of inter-tier liquid cooling techniques to avoid hotspots scenario, as they remove the heat directly from the inner layers of the 3D-MPSoC (cf. Figure 1). In addition to that, liquid cooling policies provides a value of undone workload that is less than 1% of the overall executed workload. However, air cooled polices provide values ranging from 24% to 31% in the case of $\mathbf{AC\_LB}$ and $\mathbf{AC\_TTDVFS}$, respectively.

Previous results show the reason why there is a need for liquid cooling for 3D-MPSoC structures. Because of the fact that we are interested in techniques that avoid hot spots while satisfying performance requirements, we restrict from now on our comparison to liquid cooling methods. The following paragraphs compare the proposed policy versus state of the art liquid cooling methods. The left graph of Figure 2(b) shows the overall energy consumption of the 3D MPSoC. It is divided here into two contributions. The first one is the one absorbed by the cooling network (pumps and valves) while the second is the energy absorbed by the MPSoC activity (switching and leakage). The simplest policy $\mathbf{LC\_LB}$ shows the highest energy consumption. The value of the cooling power here represents 24% the overall 3D MPSoC energy consumption. For this reason, $\mathbf{LC\_VF}$ [1] and $\mathbf{LC\_Fuzzy}$ [16] have been proposed to reduce the power consumption of the cooling system. We tested these policies on our experimental setup. They show a reduction in the cooling power consumption by approximately 30% and 50% respectively. The proposed technique has a cooling and an overall 3D MPSoC power consumption that is respectively 50% and 7% lower compared with $\mathbf{LC\_LB}$. If we compare our policy with $\mathbf{LC\_Fuzzy}$, we see approximately the same saving in terms of cooling power and a 3% additional saving in the overall MPSoC consumption.



(a)



(b)

Fig. 2. (a):Percentage of run-time execution where the maximum MPSoC temperature is higher than the threshold($370^{\circ}K$). The area of the hotspot is also provided as a percentage of the overall MPSoC area.(b) left graph: energy consumption of the overall system: 3D MPSoC power consumption and cooling network. Values are normalized to $\mathbf{LC\_LB}$;(b) right graph: average maximum 3D MPSoC temperature [$^{\circ}C$].

Finally, Figure 2(b) shows the average maximum 3D MPSoC temperature for all the policies under comparison. The lowest thermal profile among the compared policies is generated by the $\mathbf{LC\_LB}$. In this case the maximum MPSoC temperature has an average value of $54^{\circ}C$. $\mathbf{LC\_LB}$ and $\mathbf{LC\_Fuzzy}$ show a thermal profile having an average maximum temperature of $89^{\circ}C$ and $92^{\circ}C$, respectively. The reason is because both these systems save energy by reducing the cooling cost and by having the system working at a temperature close to the threshold set to $97^{\circ}C$. However, the proposed policy is able to save as much energy as $\mathbf{LC\_Fuzzy}$, while being able to keep the thermal profile $18^{\circ}C$ lower. The main reason is because the predictive problem formulation of the proposed method is able to satisfy performance requirements by acting in advance and this allows the policy a smoother control on the system and SAVES active power. Therefore, the 3D MPSoC thermal profile is colder and more thermally-balanced overall.

7 Conclusion

The contribution of this paper is a novel online thermal management approach that exploits the use of variable-flow liquid cooling on a 3D-MPSoC. In particular, we propose a thermal manager that uses DVFS and adjusts the liquid flow rate to meet the desired performance requirements, while minimizing the overall MPSoC energy consumption (MPSoC frequency setting and liquid cooling) and preventing hot spots. Our experimental results illustrate that our policy satisfies performance requirements, maintains the temperature below the specified threshold, while reducing cooling energy by up to

50% compared with traditional state-of-the-art liquid cooling techniques. The policy also keeps the thermal profile approximately $18°C$ lower compared with state of the art polices using liquid cooling.

References

1. Coskun, A.K., et al.: Energy-Efficient Variable-Flow Liquid Cooling in 3D Stacked Architectures. In: DATE 2010 (2010)
2. Coskun, A.K., et al.: Modeling and dynamic management of 3D multicore systems with liquid cooling. In: VLSI-SoC (2009)
3. Brunschwiler, T., et al.: Interlayer cooling potential in vertically integrated packages. Microsyst. Technol. (2008)
4. Kongetira, P., et al.: Niagara: A 32-way multithreaded SPARC processor. IEEE Micro (2005)
5. Skadron, K., et al.: Temperature-aware microarchitecture: Modeling and impl. In: TACO (2004)
6. Coskun, A.K., et al.: Modeling and dynamic management of 3D multicore systems with liquid cooling. In: VLSISOC (2009)
7. Sridhar, A., et al.: 3D-ICE: Fast compact transient thermal modeling for 3D-ICs with inter-tier liquid cooling. In: ICCAD (2010)
8. Mukherjee, R., et al.: Physical aware frequency selection for dynamic thermal management in multi-core systems. In: ICCAD (2006)
9. Bemporad, A., et al.: The explicit linear quadratic regulator for constrained systems. Automatica (2002)
10. Zanini, F., et al.: Multicore Thermal Management with Model Predictive Control. In: ECCTD (2009)
11. Donald, J., et al.: Techniques for multi-core thermal management: Classif. and new exploration. In: ISCA (2006)
12. Coskun, A.K., et al.: Proactive temperature balancing for low cost thermal management in MPSoCs. In: ICCAD (2008)
13. Coskun, A.K., et al.: Temperature management in multiprocessor SoCs using online learning. In: DAC (2008)
14. Grant, M., et al.: CVX: Matlab software for disciplined convex programming, `http://www.stanford.edu/~boyd/cvx/`
15. Zanini, F., et al.: Online Convex Optimization-Based Algorithm For Thermal Management of MPSoCs. In: GLSVLSI (2010)
16. Sabry, M.M., et al.: Fuzzy Control for Enforcing Energy Efficiency in High-Performance 3D Systems. In: ICCAD (2010)
17. Ruggiero, M., et al.: MPARM: Exploring the Multi-Processor SoC Design Space with SystemC. The Journal of VLSI Signal Processing (2005)
18. Magno, M., et al.: Adaptive power control for solar harvesting multimodal wireless smart camera. In: ICDSC (2009)
19. Bose, P.: Power-efficient microarchitectural choices at the early design stage. In: Keynote Address on PACS (2003)
20. Boyd, S., et al.: Convex Optimization. Cambridge University Press, Cambridge (2004)